BRL MR 2721

BRL

(12) NW

AD

MEMORANDUM REPORT NO. 2721

FEASIBILITY STUDY FOR A THREE-DIMENSIONAL, TIME-DEPENDENT HYDROCODE FOR INTERMEDIATE BALLISTICS APPLICATIONS

Csaba K. Zoltani

January 1977

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: BRL Memorandum Report No. 2721 — 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: Feasibility Study for a Three-Dimensional, Time-Dependent Hydrocode for Intermediate Ballistics Applications	5. TYPE OF REPORT & PERIOD COVERED: Final Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Csaba K. Zoltani	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: USA Ballistic Research Laboratory, Aberdeen proving Ground, Maryland 21005	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1W662603AH78-06
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Materiel Development & Readiness Command, 5001 Eisenhower Avenue, Alexandria, VA 22333	12. REPORT DATE: JANUARY 1977 13. NUMBER OF PAGES: 24
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Transitional Ballistics
Guns
Computer Simulation

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)* (ams)

The feasibility and economic viability of the development of a three-dimensional, time-dependent hydrocode for the calculation of the flow processes through a muzzle device of arbitrary geometry are discussed. It is shown that subject to some constraints, such a development is indeed within the state-of-the-art.

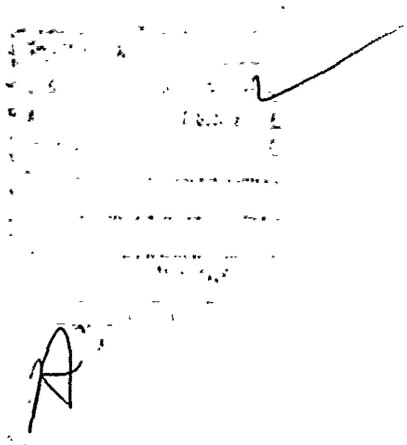
050 750

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 63 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS



I. INTRODUCTION

In this memorandum report we explore the feasibility and economic viability of the development of a three-dimensional, time-dependent hydrocode for the calculation of the flow processes through a muzzle device of arbitrary shape with a moving projectile. We assume that the working medium is a multicomponent viscid and compressible gas capable of sustaining chemical reactions. The code must be able to treat a flow field of the order of 30 calibers downstream from the muzzle, ten calibers to the rear and 20 calibers laterally. Ancillary requirements include, but are not limited to, sharp shock definition and accurate description of the projectile motion.[1]

Currently there exist several codes for muzzle flow calculations. These are the revised SAMS code of BRL and the SHELLTC of Dahlgren. The former, though the best available and giving satisfactory results, is limited in several respects: the projectile is constrained to move along the axis of symmetry of the gun tube enabling only axisymmetric muzzle devices to be modeled, and the working medium must be a one-component gas. Also, at late times, in the plane of the muzzle at several calibers from the line of fire, troublesome numerical anomalies appear.

The first serious attempt to assess the feasibility of realistic three-dimensional flow calculations is due to Gage and Mader.[2] They showed a decade ago that, indeed, given the right machine and considerable funds, such a calculation was possible though economically ahead of its time. Within the last five years appreciable increase in the speed of computers has been achieved which, coupled with the development of newer algorithms, led to the appearance of working codes for three-dimensional flow configurations. Much of the work was motivated by the need for design data for the space shuttle.[3,4,5] In addition, a number

---

[1] Zoltani, C.K., "The Intermediate Ballistic Environment of the M-16 Rifle," BRL Report No. 1860, February 1976. (AD #B010102L)

[2] Gage, W.R., Mader, C.L., "Three-Dimensional Cartesian Particle in Cell Calculations," LASL-3422, January 1966.

[3] Rizzi, A.W., Inouye, M., "Time Split Finite Volume Method for Three-Dimensional Blunt Body Flow," AIAA Journal 11, 1478-1485 (1973).

[4] Kutler, P., Sahell, L., "Three-Dimensional, Shock-on-Shock Interaction Problem," Aerodynamic Analyses Requiring Advanced Computers, Vol. I, NASA SP-347, Washington, DC, 1975, pp. 1111-1140.

[5] Kutler, P., Reinhardt, W.A., Warming, R.F., "Multishocked, Three-Dimensional Supersonic Flow Fields with Real Gas Effects," AIAA Journal 11, 657-664, (1973).

of codes were written to predict the dynamic behavior of solids[6] subject to intense loading, as well as for environmental fluid mechanics studies[7,8] and atmospheric nuclear blasts.[9] Although useful for the problem they addressed, none of these codes, or the algorithms on which these are based, were judged to take full advantage of the latest developments in hardware and software technology to warrant their adoption for muzzle flow predictions.

The Jason committee study on the "Numerical Simulation of Turbulence"[10] puts the problems requiring extensive storage into sharp focus. The authors come to the conclusion that full-scale turbulence modeling for high Reynolds number flows, of interest for many problems, will not be feasible in the foreseeable future.

The muzzle flow problem is a more modest one. In the following sections we will discuss the current state of the art as well as the expected advances in the very near future of hardware and software technologies. Then, based on an estimate of the requirements for a time-dependent, three-dimensional flow development, it will be shown that within reasonable constraints such as flow simulation is indeed feasible using existing hardware.

---

[6] Wilkins, M.L., Blum, R.E., Cronshagen, E., Grantham, P., "A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time," Lawrence Livermore Laboratory, UCRL-51574, November 1975.

[7] Hirt, C.W., Cook, J.L., "Calculating Three-Dimensional Flows Around Structures and over Rough Terrain," J. Computational Physics 10, 324-340 (1972).

[8] Hotchkiss, R.S., "The Numerical Modeling of Air Pollution Transport in Street Canyons," LA-UR-74-1427.

[9] Pracht, W.E., "Calculating Three-Dimensional Fluid Flows at all Speeds with an Eulerian-Lagrangian Computing Mesh," J. Computational Physics 17, 132-159 (1975).

[10] Case, K.M., Dyson, F.J., Frieman, E.A., Grosch, C.E., Perkins, F.W., "Numerical Simulation of Turbulence," Stanford Research Institute Report JSR-73-3, November 1973.

II. AVAILABLE HARDWARE

A. System Architecture

Computer system architecture is defined in terms of an instruction stream, a data stream and mechanisms for altering the flow of control. By judicious sequencing of the operations, such as pipelining, where an arithmetic operation is broken into a sequence of segments permitting concurrency in instruction execution, or parallel computation where the control processor directs a number of arithmetic units to perform identical operations at the same time, sizable economies in computation may be realized over third generation machines. A new code for three-dimensional, time-dependent calculations will have to be designed to take advantage of sophisticated system architectures, that is, machines which incorporate multi-processors, array processors, or associative array processors. The speedup of these machines is realized through a parallel computer organization, such as that employed in the ILLIAC IV, or pipelining found in the TI ASC and the CDC STAR.

First a review of some principles is in order. Computing speed is determined by the effective memory cycle time and the execution time of instructions in the processor. To compare different systems, it is convenient to use MIPS (million instructions per second) which represents a weighted average of execution time of a typical set of instructions characteristic for a particular class of computing tasks.

Associative array processors work on the SIMD principle: single instruction, multiple data. The ILLIAC IV is an example of this kind of machine architecture. It has a special array processor in which the PEs (processing elements) are of single construction, typically serial by bit. In addition, only one word is in the associative memory unit; that is, every word of the associative memory has its own processing unit and operations are performed concurrently by all the processors. Here all PEs obtain their instructions simultaneously from a single instruction stream. Each PE will execute this instruction stream with different data. Table 1 illustrates the characteristics of such a system. Data projected to the future in this and all succeeding tables are from Reference 11.

Table 1. Associative Array Processor Capability

	t add (μs)	t mult (μs)	Arithmetic MIPS	Search MIPS
Current	3.2	110	7.4	200
1980	2.4	55	14.0	400

[11] Turn, R., "Computers in the 1980s," Columbia University Press, New York, 1974.

Array processors are characterized by a large array, typically 64 or more, of processing elements controlled by a single control unit so that a single instruction controls the operation of all PEs. An example is the ILLIAC IV, where each processing element is capable of 2 MIPS.

A more advanced concept is a multiprocessor working on the MIMD principle (multiple instruction, multiple data), such as is incorporated in the experimental CDC 8600 machine. There the processors share a common memory but several modes of operation are possible. Separate, independent computing tasks may be performed, on different parts of the same computing task or a mixture of tasks provided by the operating system running in a multi-programmed mode.[12] See Table 2.

Table 2. Multiprocessor System Characteristics

	t add (ns)	t mult (ns)	MIPS N=1	MIPS N=4
Current	7.09-9.5	95-115	17.2-19.8	60-70
Current	3.5-6.0	50-85	27.8-41.6	97-145

Here N=1 refers to a single processor while N=4 indicates a four-processor system.

By 1985, Turn[11] expects pipelined uniprocessors to be capable of a processing rate of 650 MIPS, and a four-unit multiprocessor should be able to do 240 MIPS. Four-megabit memories should be accessible in 30 ns and 10^{10} -10^{13} bit holographic memories in a few microseconds. Bandwidth for data communication for magnetic recording should be 0.1 to 20 MHz, while laser beam recording will approach the range of 0.2 to 200 MHz.

We have seen that computers using multiprocessors are able to execute an addition or multiplication of the order of a thousand times faster than currently used associative array processors. By way of comparison, Tables 3-5 give the characteristics of the most advanced hardware available at the time of this writing. It is significant that while the CDC 7600 and the IBM 160/195 are capable of 10 MIPS the CDC STAR-100 does three times and the ILLIAC IV five times better if optimally programmed. A claim of 200 MIPS is made for the four-pipe ASC while CRAY-1 is designed to run five times faster than CDC 7600.

---

12 Feustel, E.A., Jensen, C.A., McMahon, F.H., "Future Trends in Computer Hardware," Proceedings, AIAA Computational Fluid Dynamics Conference, Palm Springs, CA, 1973, pp. 1-7

Let us take a closer look at the basic operating principles of the four computers which should be considered as potential candidates for running a three-dimensional hydrocode.

The ILLIAC IV is the only truly parallel machine in operation today. It is built around a control unit (CU) which decodes instructions, fetches operands from memory, initiates instructions and stores results. Connected to the control unit are 64 processing elements, each a computer in its own right, with a 2048-bit semi-conductor memory. This prevents delays caused by PE's referencing or altering the same memory location and also shortens the execution time in that the distance that the data must travel is reduced.

The CU may turn any of the PE's off, or the PE can turn itself off. Also by use of the registers, each PE may use a different memory location for a given memory operation, giving the PE a low degree of independence.

To efficiently utilize the ILLIAC IV, the algorithm must be as parallel as possible and the data must be stored in such a manner that data writing and calculations are done in parallel.

The great drawback, and the fact which eliminates this machine from serious contention, is that if not all 64 PE's are kept fully utilized, serious degradation of efficiency is observed. Only rarely does one encounter situations where the number of grid points turns out to be a multiple of eight.

The CDC STAR has an architecture which is partially pipeline. While the instruction processor is not a pipeline, it has three pipeline AU's. The minor cycle time per 64-bit element is 40 ns.

A distinct advantage of the STAR is its broad range of instructions and languages and large core, consisting of 4 million 32-bit words with a memory cycle time of 1100 ns. The STAR has 4 memory buses, each of which can fetch 20 64-bit words into a read buffer which services the three pipes. It takes one minor cycle to communicate between the buffer and the memory.

The STAR vector instructions are only one loop deep and a vector instruction is specified by a base address and a length. The vectors operate only on contiguous memory locations in the forward direction. In addition, there are two other vector features, the control vector and the sparse vector. Each bit of the control vector corresponds to an element in the vector operation; that is, if the i-th bit is on, the i-th result is calculated and stored into memory. However, if the i-th bit is off, the result is calculated but not stored into memory. For sparse arrays, the positional significance of each element is preserved by carrying along an order vector which locates the non-zero elements.

To take full advantage of the features of STAR, programs should be written in assembly language. This machine is especially well suited for problems where the vector contains a large number of elements.

The ASC (Advanced Scientific Computer) is built around two processors: the central processor (CP) and the peripheral processor (PP) and a large semiconductor memory containing up to 16 million 32-bit words with a memory cycle of 160 ns. It is of pipeline construction with the instruction processing unit (IPU) having four levels and the AU eight levels. Vector instructions and memory buffers have been developed to keep the pipeline full.

To minimize the delays when data fetches are executed from non-contiguous memory locations, there is an LLA (load look ahead) and a PBC (prepare to branch) instruction. The LLA defines the beginning and length of a loop in assembly code with contiguous fetching suppressed until a branch is taken outside the loop, making the code for the top of the loop available without a memory fetch. The PB is placed ahead of a branch in an instruction stream, resulting in the filling of a buffer instead of the next contiguous octet.

The most novel feature of the ASC is the handling of vector instructions. This entails a subroutine type of a call where the desired instruction and the vector parameters are specified. The machine comes with a compiler which can optimize existing FORTRAN code for pipeline processing.

Short vector specification or nonstructured programming can seriously degrade the performances of this machine. The greatest shortcoming of the ASC, however, is the short word length of 32 bits, necessitating double precision mode of operation for most scientific problems.

The technologically most advanced machine is the CRAY-1.[13] Its innovative features include 1 M 64-bit words of 50 ns bipolar LSI random access memory, chaining and the use of register-to-register vector instructions. Also, it incorporates 12 fully segmented functional units, allowing for a high degree of concurrency.

CRAY-1 uses short vectors of 64 words in length. Longer vectors are processed in segments, called vector loops. Each pass through the loop processes a 64-word segment of the vector. Once the program is inside the loop, the machine optimizes the processing by exploiting chaining and the twelve independent functional units to read, execute and return to memory the result.

---

[13] An Introduction to the CRAY-1 Computer, Cray Research, Inc., Chippewa Falls, WI, 1975.

Upon a vector instruction, a result register is reserved based on the number of clock periods determined by the vector length and functional unit time. This allows the final operand pair to be processed by the functional unit and the corresponding result to be transmitted to a result register. This way a result register becomes the operand register of the next instruction. In chaining, the succeeding instruction is issued as soon as the first result arrives for use as an operand.

Table 3

Characteristics of Currently Used Computers for Hydro Calculations

Computer	Storage Size	Storage Cycle Time (μs)	Add Time (μs)	On-Line Storage Capacity (Words)
CDC 7600	64K (small core)	0.3	0.03	80M
	500K (large core)	1.8	0.10	on disk
CDC Cyber 73	128K (small core)	1.0	1.1	96M on disk
CDC STAR	1M	0.040	1.76	
IBM 360-195	256K	0.810	0.0540	
ASC	0.5-8M	0.160		
ILLIAC IV	0.1-2M	0.188	0.012	10^7 by tes

The main advantage of the CRAY-1 is its high speed, large core and flexible handling of vectors. It is the most suitable machine for three-dimensional hydro calculations.

Table 4. On-Line Storage Devices

Device	Capability (bits)
CDC 821	0.0072×10^{12}
CDC 844	0.0028×10^{12}
IBM 1360	1.0000×10^{12}

Table 5. Characteristics of the Most Advanced Hardware Available*

	STAR-100	ASC-4 pipe	CRAY-1
MAIN STORAGE			
word size	64	32	64
max number of words	2^{20}	2^{24}	2^{22}
R/W cycle (n sec)	1000	160	48
interleave	32	8	16
COMPUTATIONAL UNIT			
cycle time	40 ns	80 ns	12 ns
M ↔ CPU parallel ch.	8	5	1
max rate (M words/s)	200	400	333
operand size	8,32,64	16,32,64	64
vector registers	buffers	buffers	512
max vector length	65535 words	-	64 words
scalar rate (CDC 7600=1)	0.25	0.27	2.5
I/O			
channels	4-12	2	12-I,12-O
bandwidth (1 ch)	0.62 M words/s	7.3	10
Advantages	Can do fast sparse vector calc.	Well organized	Fast cycle time, good scalar, vector speeds
Disadvantages	Vector start-up time is long Poor scalar speed Vectors must be consecutive	Slow cycle time	Limited memory access paths

* Supercomputers, such as PEPE (parallel element processing ensemble), built by Burroughs Corp. for the System Development Corp. incorporating distributed logic technology and designed for ballistic missile defense, will not be considered here. Though PEPE is capable of 800 MIPS as compared to about 10 MIPS of the CDC 7600 serial processor, this computer is not available for muzzle flow simulation.

B. Mass Memory Devices

Four types of memory systems are available. The mass bulk memory has a very large storage capacity, of the order of 10^7 bits, but is burdened by a relatively slow access time. Main random access memories, on the other hand, hold around 10^6 bits and are relatively fast with cycle times between 0.5 and 1.0 μs. Buffer memories are fast, featuring less than 100 ns cycle times (and are typically restricted in size to 10^5 bits). Finally, special-purpose memories provide high-speed access but they are designed for read only or mostly read only.

The Tables 6 and 7 below summarize commonly used memory characteristics.

Table 6. Mass Memory Characteristics[11]

Type	Access Time	Capacity (bits)	Transfer rate (M bits/s)
Disk	30000-75000 μs	10^8-10^9	6
Plated Wire	1-2 μs	10^8	10
Laser memory	1-10 μs	10^9-10^{12}	10
Bubble memory	2 μs	10^8-10^9	4-10

Table 7. Random Access Memory Characteristics[11]

	Read/Write (ns)
MAIN MEMORIES:	
Current	100
1980	50
BUFFER MEMORIES:	
Current	20
1980	10

As will become clear in the discussion to follow, a speedup in memory access time, Table 8, would be highly desirable. Charge-Coupled Devices[14] (CCDs) offer some improvement over currently available systems. Although they can store less than a standard disk (10^9 bits) can, i.e. 10^7-10^8 bits, they have an access time of between 100 μs and 10 ms.

Table 8. Typical Memory Access Time

Type	Time (s)
Core	10^{-7} - 10^{-6}
CCD	10^{-5} - 10^{-2}
Drum	10^{-2} - 10^{-1}
Disk	10^{-2} - 10^{-1}

Thus the CCDs, based on access times, place somewhere between main and conventional auxiliary memory but at an appreciably lower cost. Development work on magnetic bubble memories indicates that they would be substantially slower than CCDs.

Typical memory sizes of fourth generation machines are listed in Table 9. It is important to note that none of these machines is capable of holding more than 4M 60 bit-words.

Table 9. Memory Size

Machine	Memory Size (M words)	Comments
CRAY 1	0.5-1	being tested at LASL
CRAY 2	0.5-4	projected
STAR 1	0.5-1	has encountered problems
STAR 2	0.5-4	projected
ILLIAC IV	0.1-2	in operation
ASC	0.5-8	several in use, but has 32-bit word length only.

[14] Panigrahi, G., "Charge Coupled Memories for Computer Systems," Computer 9, 33-41 (1976).

To compute at maximum speed, the memory must have sufficient bandwidth to supply the arithmetic unit with operands as fast as they are needed. To increase the memory bandwidth, "interleaving" is used. This entails dividing the memory into subunits, i.e. creating a set of independently operating modules. This enables the machine to comply with multiple memory requests simultaneously, in effect creating a broader bandwidth.

C. Peripheral Devices

No appreciable improvements are expected[11] in card reader rates over the already achievable 2000 cards per minute. On the output end, non-impact printing techniques, including ink jet, electrostatic, and electro-optical, will increase the data retrieval speed. Graphics capability, as described in References 15 and 16, is essential for the evaluation of computed data for three-dimensional flows. Graphics packages are commercially available and adequate and will not be further discussed here.

D. Appraisal of the State-of-the-Art

The foregoing discussion suggests that the rate of data transfer to and from memory is the weakest link of current hardware. Although rotating drums, such as those used with the ILLIAC IV, can transfer 50×10^6 words/second, the memory can process an order of magnitude more information in the same time frame so that the system is not fully utilized. Therefore, when faster backing stores become available, they in turn will increase the overall computational power.

III. SOFTWARE

A. Background

Computation is a process of performing operations, also called mappings, as specified by instructions on a set of data. Presently used algorithms were devised for serial machines and therefore do not and cannot exploit the efficiencies offered by parallel machine architectures. As a matter of fact, vector-to-scalar machine speed ratios can be so large that even a small scalar content in a vectorized code can pose a serious degradation in performance. Thus scalar coding for a vector processor appreciably slows down a computation.

---

[15] Grantham, P., Cronshagen, E., "Computer Programs for Simulating Physical Phenomena in Three Space Dimensions and Time," Proceedings, CUBE Symposium, USERDA, Washington, DC, 1975.

[16] "Applications of Computer Graphics in Engineering," NASA SP-390, Washington, DC, 1975.

Some algorithms are unsuitable for array processors. These include, but are not limited to, finite element methods based on a mesh of an irregular topology and semi-Lagrangian schemes with fluctuating nearest neighbor relations. Also, Monte-Carlo and implicit differencing schemes yield shorter vectors than explicit differencing schemes, causing problems due to long start-up time for vector operations.

The fourth generation of computers are built around the concepts of parallel or pipelined architectures. While in a sequential machine operations are performed in a rigid sequence, arbitrary sequencing permits the number of operations to be performed concurrently, leading to an increase in the speed of the calculation. The major difference between parallel and vector architectures relates to start-up time penalties on the vector processor.

Some algorithms, on the other hand, are ideally suited for parallel program organization. These include, but are not limited to, fast Fourier transformers, matrix manipulation, and solutions of recursive problems. Miranker[17] gives a comprehensive survey of available methods.

Vectors should be several hundred elements long to keep start-up time penalities below ten percent of the total operation time on these machines. We recall that start-up time is the time required to initiate a vector instruction.

The importance of good coding practice cannot be overemphasized. For scalar work, the limit on transfer rates is dictated by the nature of the scalar instruction set, i.e., register to register, register to memory, bandwidth of associated buses, and machine cycle per instruction issue. In vector machines, like the CDC STAR, the computation time for a code that makes the machine work well to a code that doesn't can be 5 to 1. For example, a good FORTRAN program can get 30% or more of the 7600's potential.[12]

There are numerous programming devices which can speed up the running of a program. An example is the use of stack loops, where vector instructions are emulated by taking advantage of the CPU structure of the 7600.

B. Parallel Program Organization

In order to utilize the parallelism of the machine, the FORTRAN used must be extended by the addition of explicit vector arithmetic statements which then can be compiled into vector instructions on array processors. (Note that on serial machines vector statements are compiled as loops). The analogue of inner loops of serial algorithms is a

[17] Miranker, W.L., "A Survey of Parallelism in Numerical Analysis." SIAM Review 13, 524-547 (1971).

collection of vector operations in which the loop index becomes the vector index. The longer the loop, the more efficiently the vector operation will be carried out. Of course, since the loop must be executable in any order, operations such as the inversion of tri-diagonal matrices, where each step of the calculation depends on the results of the previous step, cannot be implemented efficiently.

The general rule for implicit numerical algorithms is that the innermost loop should be explicit; i.e., each step in an implicit sweep should be executed for all elements in an explicit inner loop. This will also require additional storage for working vectors.

For explicit marching schemes the inner loop should be in the longest mesh direction and boundary conditions should not be handled as a special algorithm, but as part of the general interior calculation.

In programming array processors, the separate PEs create serious complications in arriving at an efficient storage allocation scheme.

C. Current Three-Dimensional Codes

Three installations, the Los Alamos Scientific Laboratory, NASA-Ames and the Lawrence Livermore Laboratory, have working codes for three-dimensional flow simulation. They address a variety of problems ranging from steady, incompressible, inviscid to the unsteady, supersonic, full Navier-Stokes equations. The algorithms used may be broken down into two general classes: references 7, 8 are employing variants of the particle in cell methodology and the other is using the MacCormack scheme in various formulations (references 3, 4). An interesting development in conjunction with the latter was the emergence of the CFD programming language[18] used on the ILLIAC IV.

The results reported are encouraging, but in general the methods are too problem-oriented to allow a direct application to the muzzle flow problem. This is especially true for problems run on the ILLIAC IV. The speedup over the CDC 6600 is impressive; for example, for a single material code approximately 0.5 ms/zone/cycle was required, which is a factor-of-three improvement. One is restricted though to certain multiples of grid points for maximum machine efficiency and the time saving over conventional methods is strongly dependent on the algorithm used. In general, it has been found that running times can be reduced if internal checks and jumps are held to a minimum. This is so because on parallel machines internal checks are usually made by only one of the processors while the others are idling. Also, indexing calculations should be avoided since data is stored in a continuous array, and finally data blocks are preferred; i.e., instead of separate

[18] Stevens, K.G., Jr., "CFD-A FORTRAN like Language for the ILLIAC IV." Paper presented at the NASA/ACM Conference: Programming Langugages and Compilers for Parallel and Vector Machines. Goddard Institute for Space Studies, 1975.

arrays for each parameter, the parameters are stored for a given zone in consecutive locations in memory.

Most three-dimensional calculations reported to date have been run for a small number of grid points. Patankar[19] reports on the simulation of flow in a cavity created by the movement of a plate over the free surface. For a grid of 8 x 8 x 8 and at a Reynolds number of 100, the calculation took 30 seconds on the CDC 6600. The flow pattern and radiation determination in a gas turbine combustion chamber, where turbulence and combustion were included in the model, for a grid of 7 x 7 x 7 increased to 60 seconds on the same machine.

Gentry, et al[20], using the BAAL code[9], calculated the time-dependent surface pressure history generated on the surface of a rectangular obstacle by the passage of a weak shock. The full Navier-Stokes equations were used for a grid of 8381 cells with 27 variables per cell. The flow simulation took 40 minutes on the CDC 7600 and necessitated both basic and large core memory. Thus, one can see that for a realistic number of grid points, using conventional methods, the running of these problems becomes uneconomical. As a rule of thumb they state that "all other things being equal," the cost of 3D numerical calculations rises as Δx^{-4} where Δx is the mesh size.

IV. ESTIMATE OF REQUIRED CAPABILITY FOR MUZZLE FLOW CALCULATIONS

Based on our experience with two-dimensional time-dependent simulation of the intermediate ballistic region[1], for acceptable resolution of the salient flow details, 150 mesh points in the axial and 60 points in the radial direction are required. In three dimensions, the aximuthal co-ordinate, assuming the worst case of complete nonsymmetry about the tube axis, will add another 300 points, requiring a total of 2.7×10^6 mesh points. Three velocity components, two thermodynamic quantities, and conservatively estimated five chemical species give 10 variables per cell, requiring in the neighborhood of 30 M words of storage. Co-ordinate stretching, to reduce the number of grid points, would not be advisable due to the loss of shock and flow definition which is of paramount interest.

To estimate the approximate running time we adopt the methodology of reference 10. We assume that a fast algorithm was chosen to solve the Navier-Stokes equation with the appropriate boundary and initial conditions. With N mesh points in each of the spatial directions, $10\ N^3$ variables will have to be determined for each of the 10^3 sweeps of the computational grid.

---

[19] Patankar, S.V., "Numerical Prediction of Three-Dimensional Flows." Studies in Convection, Volume 1. Ed. B.E. Launder, Academic Press, London 1975, pp. 1-78.

[20] Gentry, R.A., Stein, L.R., Hirt, C.W., "Three-Dimensional Computer Analysis of Shock Loads on a Simple Structure." BRL Contract Report No. 219, March 1975. (AD #B003208L)

A rough estimate of the operations count of the updating of each grid point may be made by assuming that the main part of the calculation is the determination of the pressure. This is done by applying a fast Fourier transform to the Poisson equation. In addition, we must include in the total count the manipulation needed for advancing the velocity to the new time level.[21]

$$2N^3 \ell n_2 N^3 \text{ additions and}$$

$$N^3 \ell n_2 N^3 \text{ multiplications}$$

are needed for the FFT and approximately 300 N^3 equivalent additions for the other operations per sweep. If one assumes that two additions take the same order of time as one multiplication, for 10^3 sweeps, one will then need approximately

$$1000[4 N^3 \ell n_2 N^3 + 300 N^3]$$

operations per sweep. Total time for the calculation is then

$$T = \text{addition time} \times 1000 \ [4N^3 \ell n_2 N^3 + 300 N^3]$$

This estimate is very conservative. On parallel or pipeline machines an improvement of several orders of magnitude is expected.

At this juncture it is useful to recall the capability of the best of the current generation of computers. Typical add and multiplication times are 100 ns and 200 ns respectively, with a fast memory holding around 0.5×10^6 64-bit words. Disk storage of 10×10^6 64 bit words is not uncommon. Transfer rate from disk to working array is 10^7 words per second. However, disks are subdivided into bands, each band containing 300 pages, each with 10^3 64-bit words. One can transfer one page at a time which takes around 130 microseconds.

The constraints of the problem are then the hardware characteristics, the number of grid points and the operations count needed to accomplish the computational task.

For 10^3 sweeps of a grid of 10^6 points, with two variables per point, the computation would take close to 20 hours on a machine with the capability of the CRAY-1. Of this time, approximately one hour would be spent transfering data from core to backing stores. Larger grid sizes, requiring the transfer of data from central core to disks and back again, are extremely time consuming and too costly at the time of this writing.

[21] Brigham, E.O., "The Fast Fourier Transform." Prentice Hall, Englewood Cliffs, NJ, 1974.

V. RECOMMENDATIONS

It is recommended that a three-dimensional, time-dependent, compressible, viscid, multimaterial hydrocode be developed for muzzle flow calculations. Modularized, top-down programming methods should be used with the option of extending the data base as increased needs dictate. Also, the algorithm to be developed should take advantage of the efficiencies offered by multiprocessor machine architectures. Concurrently, plotting and on-line graphical display packages should be purchased to facilitate data reduction.

Initially, the size of the grid should be such that the storage requirements for the code would not exceed one million words. This constraint has at least two advantages: one, the CRAY-1, the most advanced scientific computer now available, could be used without the need of resorting to extensive data transfer in and out of memory while the calculation is proceeding; second, running times would be held to within reasonable limits. At the conclusion of this code development, more powerful machines, possibly multiprocessors, will become available, allowing a greater number of mesh points and more variables, such as those describing complicated chemical reactions, to be treated in a routine manner. By then, estimated to be the early 1980's, the determination of the complete flow picture, regardless of the geometry or other complicating factors, from shot ejection up to and including the time that the projectile leaves the intermediate ballistic range, will become economically feasible. Projection of programming effort required for such a code development is hazardous at best. But based on discussions with scientists at LASL and elsewhere, it appears reasonable to conclude that two senior scientists, assisted by three top-flight programmers, could accomplish the task in three years.

ACKNOWLEDGEMENTS

The author would like to thank Mr. C. L. Mader of LASL and Dr. W. Jones of NASA-AMES for useful discussions in the course of the preparation of this manuscript.

REFERENCES

1. Zoltani, C.K., "The Intermediate Ballistic Environment of the M-16 Rifle," BRL Report No. 1860, February 1976. (AD #B010102L)

2. Gage, W.R., Mader, C.L., "Three-Dimensional Cartesian Particle in Cell Calculations," LASL-3422, January 1966.

3. Rizzi, A.W., Inouye, M., "Time Split Finite Volume Method for Three-Dimensional Blunt Body Flow," AIAA Journal 11, 1478-1485 (1973)

4. Kutler, P., Sahell, L., "Three-Dimensional, Shock-on-Shock Interaction Problem," Aerodynamic Analyses Requiring Advanced Computers, Vol. I, NASA SP-347, Washington, DC, 1975, pp 1111-1140

5. Kutler, P., Reinhardt, W.A., Warming, R.F., "Multishocked, Three-Dimensional Supersonic Flow Fields with Real Gas Effects," AIAA Journal 11, 657-664, (1973).

6. Wilkins, M.L., Blum, R.E., Cronshagen, E., Grantham, P., "A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time," Lawrence Livermore Laboratory, UCRL-51574, November 1975.

7. Hirt, C.W., Cook, J.L., "Calculating Three-Dimensional Flows Around Structures and over Rough Terrain," J. Computational Physics 10, 324-340 (1972).

8. Hotchkiss, R.S., "The Numerical Modeling of Air Pollution Transport in Street Canyons," LA-UR-74-1427.

9. Pracht, W.E., "Calculating Three-Dimensional Fluid Flows at all Speeds with an Eulerian-Lagrangian Computing Mesh," J. Computational Physics 17, 132-159 (1975).

10. Case K.M., Dyson, F.J., Frieman, E.A., Grosch, E.C. Perkins, F.W., "Numerical Simulation of Turbulence," Stanford Research Institute Report JSR-73-3, November 1973.

11. Turn, R., "Computers in the 1980s," Columbia University Press, New York, 1974.

12. Feustel, E.A., Jensen, C.A., McMahon, F.H., "Future Trends in Computer Hardware," Proceedings, AIAA Computational Fluid Dynamics Conference, Palm Springs, CA, 1973, pp. 1-7.

13. "An Introduction to the CRAY-1 Computer," Cray Research, Inc., Chippewa Falls, WI, 1975.

REFERENCES (CONT'D)

14. Panigrahi, G.,"Charge Coupled Memories for Computer Systems," Computer 9, 33-41 (1976).

15. Grantham, P., Cronshager, E., "Computer Programs for Simulating Physical Phenomena in Three Space Dimensions and Time," Proceedings, CUBE Symposium, USERDA, Washington, DC, 1975.

16. "Applications of Computer Graphics in Engineering," NASA SP-390, Washington, DC, 1975.

17. Miranker, W.L., "A Survey of Parallelism in Numerical Analysis." SIAM Review 13, 524-547 (1971).

18. Stevens, K.G., Jr., "CFD-A FORTRAN Like Language for the ILLIAC IV." Paper presented at the NASA/ACM Conference: Programming Languages and Compilers by Parallel and Vector Machines. Goddard Institute for Space Studies, 1975.

19. Patankar, S.V., "Numerical Prediction of Three-Dimensional Flows." Studies in Convection, Volume 1. Ed. B.E. Launder, Academic Press, London 1975, pp. 1-78.

20. Gentry, R.A., Stein, L.R., Hirt, C.W., "Three-Dimensional Computer Analysis of Shock Loads on a Simple Structure." BRL Contract Report No. 219, March 1975. (AD #B003208L)

21. Brigham, E.O.,"The Fast Fourier Transform." Prentice Hall, Englewood Cliffs, NJ, 1974.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Systems Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703
6	Commander US Army Missile Command ATTN: DRSMI-R DRSMI-RDK Mr. R. Becht (4 cys) Mr. R. Deep Redstone Arsenal, AL 35809
1	Commander US Army Tank-Automotive Development Command ATTN: DRDTA-RWL Warren, MI 48090

No. of Copies	Organization
2	Commander US Army Mobility Equipment Research & Development Command ATTN: Tech Docu Cen, Bldg. 315 DRSME-RZT Fort Belvoir, VA 22060
2	Commander US Army Armament Command ATTN; P. Ehle Technical Lib Rock Island, IL 61202
2	Commander US Army Armament Command ATTN: Rodman Laboratories, S. Thompson S. Burley Rock Island, IL 61202
6	Commander US Army Frankford Arsenal ATTN: Mr. T. Boldt SARFA-U2100 Mr. J. Mitchell SARFA-U3100, S. Fulton SARFA-U3300 Mr. S. Hirshman Mr. A. Cianciosi L4100-150-2 Mr. C. Sleischer, Jr. Philadelphia, PA 19137
5	Commander US Army Picatinny Arsenal ATTN: SARPA-DR-D, S. Wasserman SARPA-DR-V, Mr. A. Loeb Mr. D. Mertz Mr. E. Friedman SARPA-D, Mr. Lindner Dover, NJ 07801

DISTRIBUTION LIST (Cont'd)

No. of Copies	Organization
5	Commander US Army Picatinny Arsenal ATTN: SARPA-V, E. Walbrecht Mr. S. Verner SARPA-VE, Dr. Kaufman SARPA-FR-M-MA Mr. E. Barrieres Mr. G. Demitrack Dover, NJ 07801
4	Commander US Army Watervliet Arsenal ATTN: Tech Lib SARWV-PDR-S Dr. F. Sautter SARWV-RDR-AMM Dr. J. Zweig SARWV-RDR Dr. T. Simkins Watervliet, NY 12189
1	Commander US Army Harry Diamond Laboratories ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SA White Sands Missile Range NM 88002
1	Commander US Army Research Office ATTN: CRD-AA-EH P.O. Box 12211 Research Triangle Park, NC 27709
1	Commander US Army Ballistic Missile Defense Systems Command Huntsville, AL 35804
1	Director US Army Advanced BMD Technology Center P.O. Box 1500 Huntsville, AL 35809
3	Commander US Naval Air Systems Command ATTN: AIR-604 Washington, DC 20360
3	Commander US Naval Ordnance Systems Command ATTN: ORD-9132 Washington, DC 20360
2	Commander Naval Sea Systems Command ATTN: SEA-03513, Mr. Lionel Pasiuk Research of Technology Directorate Materials & Mechanics Division Washington, DC 20014
1	Los Alamos Scientific Laboratory ATTN: Mr. Charles L. Mader Los Alamos, NM 87545
	Aberdeen Proving Ground
	Marine Corps Ln Ofc Dir, USAMSAA